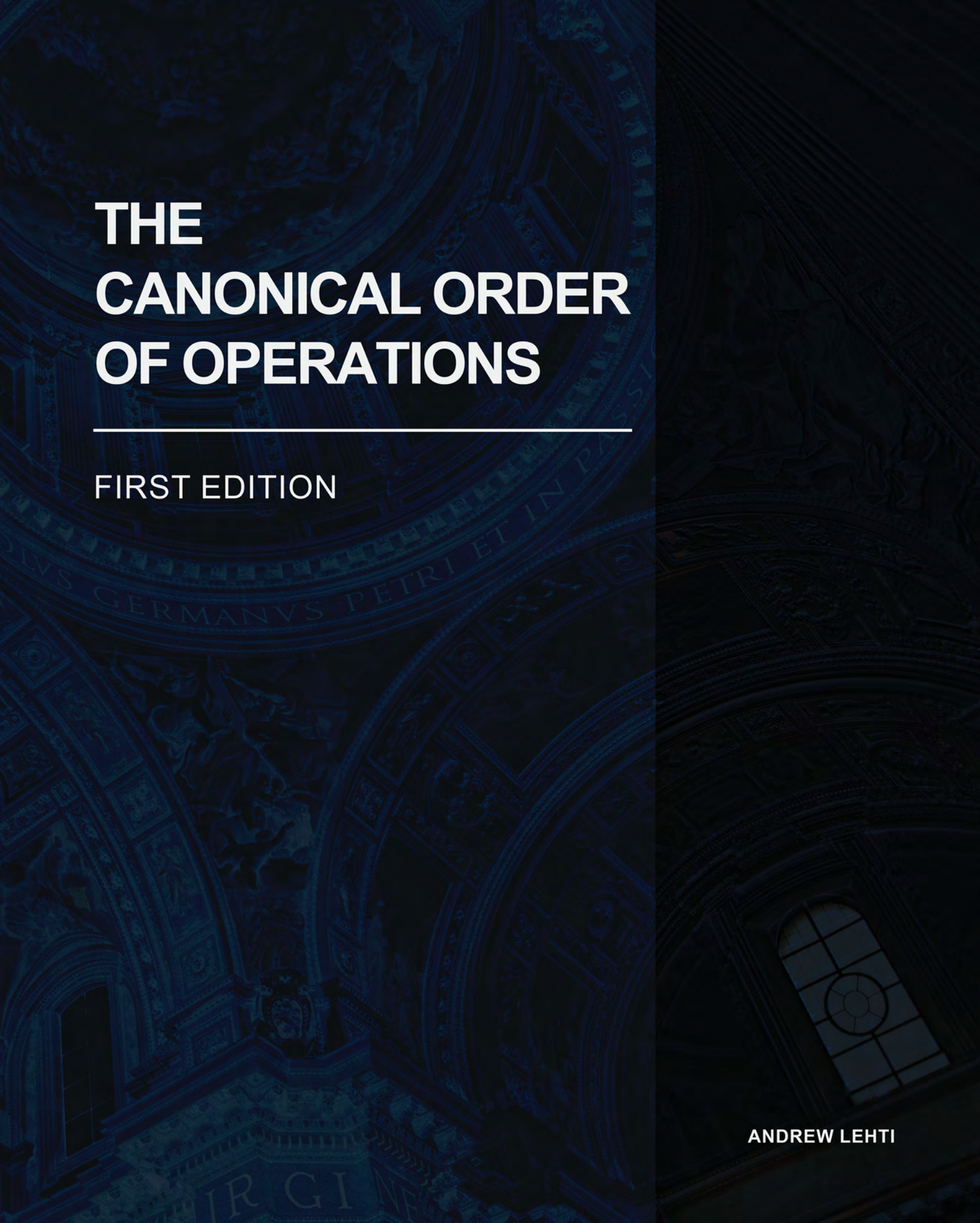

THE CANONICAL ORDER OF OPERATIONS

FIRST EDITION

ANDREW LEHTI

The Canonical Order of Operations

First Edition

Andrew Lehti
Official Manuscript

© Andrew Lehti, 2021–2025.
Licensed under Creative Commons Attribution-NonCommercial 4.0 International
CC BY-NC 4.0

ISBN: 9798292210788

Author Preface

I am not a product of formal instruction, but of relentless curiosity. My name is Andrew Lehti, and I am a polymath autodidact who has devoted over 30,000 hours to studying human cognition—not through rote memorization, but through exploration, synthesis, and reconstruction. Where others found leisure in entertainment, I found purpose in patterns. Mathematics, cognitive psychology, science, linguistics, and the histories that shaped them have formed the core of my life's investigation.

This work arises from a simple premise: if mathematics is to be the language of the universe, then its grammar must be consistent. The Canonical Order of Operations is the result of observing a deep, long-standing inconsistency in that grammar—one that has gone largely unquestioned not due to its truth, but due to its ubiquity. It is the product of thinking outside the institution, yet rigorously within logic.

I did not set out to challenge tradition for its own sake. I set out to understand, and in doing so, I found contradiction where there should have been clarity. This book is not a rebellion. It is a repair.

Lehti, Andrew (2024). ***Selective-Mindedness: An Introduction and the Illusion of Open-Mindedness****. figshare. Journal contribution. https://doi.org/10.6084/m9.figshare.27642519*

Lehti, Andrew (2024). ***Cognitive Impasse and the Puppet Master of Society: A Framework of Mental Rigidity****. figshare. Journal contribution. https://doi.org/10.6084/m9.figshare.28014626*

Lehti, Andrew (2024). ***Standardized Obedience: The Suppression of Critical Thinking, Innovation, and Creativity in Worldwide Conformity-Driven Education Systems****. figshare. Journal contribution. https://doi.org/10.6084/m9.figshare.28015913*

Lehti, Andrew (2024). ***The Cycle of Inferiority and Superiority: From Imposition to Projection and Self-Perpetuation****. figshare. Journal contribution. https://doi.org/10.6084/m9.figshare.28013819*

Lehti, Andrew (2024). ***Extrapolative Trial by Error****. figshare. Journal contribution. https://doi.org/10.6084/m9.figshare.27643080*

Lehti, Andrew (2024). ***Volume Six: Cognitive Defensiveness: Infamicate****. figshare. Journal contribution. https://doi.org/10.6084/m9.figshare.27098722*

A Note from the Author

Most people avoid cognitive discomfort by rejecting, ridiculing, or shutting down the information input. But when you choose to ford that tension—the headache, cognitive discomfort, feelings of guilt and impending doom—that's when growth happens.

The reason for discomfort and pain is because it's training parts of the brain which go untrained in society and most educations—no matter the level. Without proper training, most people will never touch anything that's different; not because they can't, but because they won't. The real challenge is learning that 'knowing' is not understanding.

Very few people will bleed for the clarity of understanding. That isn't because they're lazy, dull, or unintelligent but because they're not trained to handle a cognitive impasse, which includes the pain of dissonance. The pain of dissonance is easily mistaken for guilt, and the feeling of impending doom, or even cognitive overload; for some, it can feel like one's soul is at risk of eternal damnation. This feeling isn't unlike the stages of grief. When these feelings arise, the best thing to do is to take a short nap once things stop making sense.

In order to grow, we must be uncomfortable, and we must endure the headaches and discomfort of dissonance in order to change and progress into new personal and societal frontiers.

It doesn't have to feel like or be a struggle. It has to be a habit; one which demands exploring the uncomfortableness to find truth. To power through these feelings is no easy task, but it's what separates the thinkers from those who think they thought.

There's nothing elite about being a thinker, no praise nor glory, but there is personal satisfaction from the exploration and discovery of new insights, information, and reasoning. You'll find that most personal discoveries have already been discovered, even if the congruence isn't there. However, you may always provide new evidence or a new avenue of thought to move us forward. You may find new evidence which causes you to disprove yourself, and you should always aim to do so in order to be truly objective.

True understanding arises not from knowing, but from enduring the discomfort of not knowing. This path requires forming a habit of confronting discomfort—not with struggle, but with intention. Growth demands mental endurance.

*The following section **presents** a real-world mathematical application. While this book challenges convention, the mathematics shown below do not. Instead, it reflects standard practice and my technical competence.*

Author Mathematical Works

The Polyhedral Index Partition

Author: Andrew Lehti
DOI: 10.6084/m9.figshare.27642783

The Polyhedral Index Partition is a deterministic bijection that converts any finite coordinate tuple into a single scalar integer, and reconstructs the tuple from that integer without loss. This is not a hash. It does not approximate. It does not collide. It is a computation-based mapping built from simplicial polytopic numbers. It scales indefinitely because it is not bounded by storage structures, tables, or probabilistic design.

I. Definitions

Let the coordinate tuple be:

$$\mathbf{x} = (x_1, x_2, \ldots, x_d)$$

Assume $x_i \in \mathbb{N}$ unless otherwise specified.

Define the **suffix sums**:

$$s_i = \sum_{k=i}^{d} x_k$$

So $s_d = x_d$, and $s_1 = x_1 + x_2 + \cdots + x_d$.

Define the **simplicial number** (simplex count) as:

$$\Delta(m, d) = \binom{m + d - 1}{d}$$

Where m is the shell magnitude and d is the dimension.

This is the point-count of a d-dimensional simplex "filled" to side-length m.

II. The Partition (Encoding)

The index is formed by summing simplex counts evaluated on cumulative structure, specifically pairing the largest sums with the highest dimensions.

Given $\mathbf{x}$, compute the suffix sums $(s_1, \ldots, s_d)$

Then define the scalar index:

$$I(\mathbf{x}) = \sum_{i=1}^{d} \Delta(s_i, d - i + 1)$$

This produces a unique integer for every tuple. No collisions are possible because the construction is bijective by polytopic shell ordering.

Example (3D):

Let: $\mathbf{x} = (a, b, c)$
Then: $s_1 = a + b + c, \quad s_2 = b + c, \quad s_3 = c$

The dimension $d = 3$. The index is calculated as:

$$I(a, b, c) = \Delta(s_1, 3) + \Delta(s_2, 2) + \Delta(s_3, 1)$$

Substituting the sums:

$$I(a, b, c) = \Delta(a + b + c, 3) + \Delta(b + c, 2) + \Delta(c, 1)$$

III. The Inverse Partition (Decoding)

Decoding is a method of shell extraction.

Given I and dimension d, recover $\mathbf{x}$ by repeatedly finding the largest shell that does not exceed the remaining index.

Let $R_d = I$. Iterate k from d down to 1:

1. Find the largest integer m_k such that:
   $\Delta(m_k, k) \leq R_k$
2. Update the remainder:
   $R_{k-1} = R_k - \Delta(m_k, k)$

After shells $(m_d, m_{d-1}, \ldots, m_1)$ are found, recover coordinates by differencing the shells:

$$x_k = m_{d-k+1} - m_{d-k+2}$$

(Note: For the final element, treating the nonexistent next shell as 0.)

This is the inverse of the partition because the forward map is constructed from nested simplex counts. The inverse simply peels those layers back from the highest dimension to the lowest.

IV. Significance

- **Generalized Cantor Pairing (Beyond 2D).** This is a direct multi-dimensional bijection, not a chained pairing hack.
- **Space Filling Without Collisions.** It provides a stable scalar index for any discrete coordinate system.
- **Compression by Structure.** Many state vectors and coordinate records can be represented as a single integer without metadata tables.

V. Optimized Code

This code takes the original Polyhedral Index Partition (PIP) to the next level by handling the most complex combinations that previously pushed the limits of feasibility: cases where the integer value spans thousands of digits while the number of partition (dimensions) splits were a low value (typically under 10-100 depending on the size of the integer), as well as cases where the integer is any size but the number of partitions (dimensions) scales from tens of thousands into the millions of dimensions. Scenarios that once required minutes or hours now resolve in seconds, with benchmark results showing the optimized implementation to be 1547.41× faster than the prior optimized (unreleased) version and 21,033.43× faster than the original (released) code, while maintaining exact output, correct ordering, and full consistency with the original PIP definition.

GitHub Benchmark Test: https://bit.ly/3NwUd6l
new: 4.742585 seconds
old: 1134.30792 seconds

```
from __future__ import annotations

from dataclasses import dataclass
from math import comb
from typing import Iterable, List, Sequence, Tuple

def polyHedralIndex(x: int, y: int) -> int:
    if y < 1:
        raise ValueError("dimension d must be >= 1")
    if x < 0:
        raise ValueError("magnitude m must be >= 0")
    return comb(x + y - 1, y)

def getInverse(n: Sequence[int], y: bool = False) -> int:
    if not n:
        raise ValueError("coords must be non-empty")
    a = list(n)
    if y:
        a = [zigzagEncode(z) for z in a]
    if any((not isinstance(v, int)) for v in a):
        raise TypeError("all coordinates must be integers")
    if any(v < 0 for v in a):
        raise ValueError("coordinates must be >= 0 (or use signed=True)")
    t = len(a)
    c = [0] * t
    s = 0
    for i in range(t - 1, -1, -1):
        s += a[i]
        c[i] = s
    k = 0
    for i in range(t):
        k += polyHedralIndex(c[i], t - i)
    return k
```

… continued on next page

```
def partition(k: int, y: int, *, signed: bool = False) -> Tuple[int, ...]:
    if not isinstance(k, int) or k < 0:
        raise ValueError("index must be an integer >= 0")
    if not isinstance(y, int) or y < 1:
        raise ValueError("dim must be an integer >= 1")
    r = k
    p: List[int] = []
    for t in range(y, 0, -1):
        x = approximate(r, t)
        p.append(x)
        r -= polyHedralIndex(x, t)
    a: List[int] = []
    for i in range(y - 1):
        a.append(p[i] - p[i + 1])
    a.append(p[-1])
    if signed:
        a = [zigzagDecode(n) for n in a]
    return tuple(a)

def approximate(n: int, t: int) -> int:
    if n < 0:
        raise ValueError("remainder r must be >= 0")
    if n == 0:
        return 0
    l, h = 0, 1
    while polyHedralIndex(h, t) <= n:
        h *= 2
    l = h // 2
    while l + 1 < h:
        m = (l + h) // 2
        if polyHedralIndex(m, t) <= n:
            l = m
        else:
            h = m
    return l

def zigzagEncode(z: int) -> int:
    if not isinstance(z, int):
        raise TypeError("z must be int")
    return 2 * z if z >= 0 else -2 * z - 1

def zigzagDecode(n: int) -> int:
    if not isinstance(n, int) or n < 0:
        raise ValueError("n must be int >= 0")
    return n // 2 if (n % 2 == 0) else -(n // 2) - 1
```

Interactive Polyhedral Indexer Web Portal: https://bit.ly/4pDdTmM

Introduction

Mathematics is often celebrated as the language of precision and logic, but beneath its apparent rigor lies a surprising truth: even the most basic operations can harbor confusion. Over centuries, subtle inconsistencies in how we interpret arithmetic expressions have crept into the foundations of education. Mnemonics like PEMDAS and BODMAS are taught as gospel, yet they frequently collapse under the weight of ambiguity in more complex expressions—especially involving negatives, roots, and exponents. What begins as clarity for the novice quickly becomes contradiction for the advanced.

This book introduces *The Canonical Order of Operations*, a logically unified framework designed to eliminate these ambiguities. Rooted in cognitive psychology, mathematics, and historical analysis, this work redefines foundational rules to restore consistency without sacrificing accessibility. It discards inherited conventions when they conflict with truth and re-centers arithmetic on coherence over tradition.

What follows is not a revision but a correction—a return to clarity through principles that hold even under stress. The Canonical Order is not just a new method; it is a resolution.

Why Rethink the Order of Operations?

Mathematics is often presented as a rigid system, immune to ambiguity. Yet, beneath this perceived certainty lies a troubling reality: mathematical operations, particularly exponentiation and root extraction, have been inconsistently defined for centuries. This inconsistency has led to the unnecessary introduction of imaginary numbers, conceptual misunderstandings in education, and a framework that prioritizes convenience over logical coherence.

The *Canonical Order of Operations* seeks to address these long-standing issues by restoring mathematical consistency. Unlike the *Standard Order*, which evolved through arbitrary conventions and historical patchwork, the Canonical Order is constructed from fundamental principles that eliminate ambiguity, ensuring that mathematical rules remain universally applicable.

By systematically revising the handling of negative numbers, roots, and exponentiation, this framework makes mathematics more intuitive, accessible, and logically sound.

Why the Canonical Order Matters

The *Canonical Order of Operations* is not a new set of rules—it is a *correction* to centuries of mathematical inconsistency. It does clarify rules by adding *the Law of Implicit Unity*. By restoring logical coherence to exponentiation and root operations, it provides a solid foundation for students, educators, and researchers alike.

This framework eliminates unnecessary reliance on imaginary numbers, clarifies the treatment of negative bases, and ensures that mathematical operations remain internally consistent.

The challenge now is adoption. While the *Standard Order* remains deeply ingrained in education, those who prioritize truth over tradition must recognize its limitations. The *Canonical Order* offers a path forward, one that prioritizes logical consistency over historical inertia.

Mathematics should not be a series of patchwork rules designed to accommodate contradictions. It should be a system of truth, built upon principles that hold in all cases. The *Canonical Order* restores this foundation, ensuring that the pursuit of mathematical understanding is free from ambiguity.

This shift's importance requires recognizing the *Standard Order of Operations'* foundational role. It is the backbone of global mathematics education, a structure whose assumptions are seldom questioned. These assumptions introduce inconsistencies that affect higher mathematics. The Standard Order merges rules of multiplication, exponentiation, and sign placement, creating conflicting results with negative numbers and roots. These contradictions are not academic curiosities; they shape the framework for interpreting equations, modeling reality, and building theories of the physical universe.

Imaginary numbers are an example. They emerged as a solution for problems the Standard Order mishandled, including roots of negative numbers. The need for imaginary numbers can be traced to flawed assumptions about operation sequencing, not an intrinsic property of mathematics. This has clouded the interpretation of many equations, adding abstraction where simplicity would suffice. Imaginary numbers have applications in signal processing or quantum theory, but their premature introduction into general arithmetic obscures instead of clarifies.

The Canonical Order offers an alternative path. It does not seek to replace the Standard Order but to exist alongside it as a defined, consistent, and unbiased framework. The Standard Order can continue as a pedagogical tool for basic notation, while the Canonical Order provides a logical system for depth, clarity, and mathematical integrity. Treating them as separate frameworks allows each to serve its purpose without contradiction.

This is more than an academic distinction. How we teach and use operations affects how we interpret data, simulate systems, and understand the laws of our universe. By disentangling the Standard Order's conflated logic and presenting the Canonical Order as a consistent alternative, this book opens a door to a more precise and honest mathematics.

Main Paper

Interpreting this index law:

$$(a^m)^n = a^{m \times n} = a^{mn}$$

$$(a)^m = (a^1)^m = a^{1 \times m} = a^m$$

The Canonical Order of Operations

Author: Andrew Lehti

Disciplines: Cognitive Psychology, Linguistics, Mathematics, and their Histories

DOI: 10.6084/m9.figshare.27661734

Date: November 14, 2024

Part of the Collection of:
Cognitive Psychology and the Education System
https://doi.org/10.6084/m9.figshare.c.7532079

Learned behaviors and cognitive biases have perpetuated trait inheritance across generations, stemming from an education system established in the fourth century CE, forming a self-sustaining structure resembling a virus causing humans to deny change.

The Semmelweis Reflex is the tendency to reject new evidence because it contradicts established beliefs. Named after Ignaz Semmelweis, who introduced handwashing to reduce infections but was dismissed for challenging medical norms. After being dismissed from his position, he faced opposition and isolation within the medical community, as his handwashing theories were ridiculed and ignored. Eventually, he was committed to a mental asylum, where he reportedly became violent. Soon after his admission, he sustained a severe wound on his hand, potentially from a beating by the staff or another violent encounter, which led to sepsis. This infection caused his death at just 47.

Updated 7/12/2025

To address common concerns about accessibility, it is assured that no paywall will be implemented on core elements of this paper. This research will remain freely accessible, as research should inherently be open to all.

Methodology

The research focuses on investigating motives, behaviors, and observations while deliberately avoiding prior exposure to academic findings utilizing my approach which is **extrapolative trial by error.** This approach minimizes anchoring biases that could influence the thought process. After completing the study, academic material is reviewed comprehensively to evaluate alignment with the findings. Typically, the findings correspond closely with established principles, with exceptions limited to nuanced insights not yet formally recognized but widely observed in society. Few contradictions arise with established norms, apart from the integration of multiple established biases into more complex, composite biases that reflect broader systemic issues.

Lehti, Andrew (2024). Extrapolative Trial by Error. Journal contribution. https://doi.org/10.6084/m9.figshare.27643080

Introduction to the Cognitive Impasse Phenomenon

The cognitive impasse is a phenomenon that fosters a self-perpetuating cycle of mental rigidity, ensuring that the frameworks governing thought remain unchanged even as they are adopted by new individuals and institutions. This persistent recycling of identical intellectual structures—wrapped in new contexts and bodies—explains why the educational system has remained fundamentally unaltered for over 1,400 years. Originating from the Roman model of memory discipline and compliance, these paradigms have spread globally, embedding themselves into the fabric of schools, colleges, universities, and academic institutions worldwide.

While the tools, technologies, and methodologies of education have evolved, the underlying framework remains steadfast, prioritizing conformity and the avoidance of error over critical thinking and adaptability. The cognitive impasse not only sustains this system but also prevents those within it from seeing its flaws clearly—particularly if they are unwilling to confront the discomfort of challenging ingrained beliefs. This resistance to change ensures that educational paradigms persist, reinforcing societal stagnation masked as progress.

Ultimatum pro Canonica Mathematica

1. The Canonical Order of Operations corrects a recurring error in calculation. This error stems from misinterpretations that have been layered into textbooks and reinforced through teaching.
2. Each generation inherits the flaw, and the system amplifies it. Overhauling the current structure is impractical, but ignoring the flaw is unacceptable.
3. This work introduces a separate framework. *The Standard Order* is the legacy model. *The Canonical Order* follows the original logic of index laws without compromise or contradiction.
4. Academics must choose: correctness or tradition.
5. *The Standard Order* protects institutional habits but creates *real-world* errors. Clinging to it is not neutrality—it is bias.
6. If the goal is truth, then false structures must be discarded. Congruence is not truth. Authority is not truth. Self-deception is not truth.
7. Truth is uncomfortable. Avoiding it for comfort turns science into dogma. Persisting with the Standard Order equates to choosing bias over clarity.

Abstract

The Canonical Order of Operations introduces a unified framework to resolve ambiguities in traditional arithmetic, particularly in handling negative values, roots, and exponents. By eliminating inconsistencies that necessitate the use of imaginary numbers and redefining the treatment of operations involving negatives, this framework preserves the integrity of real numbers without resorting to abstract constructs. Rooted in centuries of mathematical tradition, the Canonical Order replaces symbols such as the radical ($\sqrt{}$) with explicit fractional exponents, making these concepts more accessible and intuitive for learners. This paper explores the historical context, defines the Canonical Order, and demonstrates its potential to simplify mathematical education and computation while maintaining logical consistency.

History

Mathematics has long contended with the representation of roots and exponents. Traditional arithmetic often defines exponents as repeated multiplication, while roots are represented using symbols such as $\sqrt{}$ However, negative bases and fractional exponents introduced complexities that led to the development of imaginary numbers, expressed as $i = \sqrt{-1}$. These constructs, while mathematically elegant, introduced additional abstractions that often confused learners and extended the number system beyond real numbers.

Prominent mathematicians, including René Descartes and Leonhard Euler, questioned the necessity of imaginary numbers, proposing that improved handling of negatives might obviate such abstractions. Yet, due to inconsistencies in the order of operations and ambiguities surrounding the radical symbol, these ideas remained unrealized.

The Canonical Order of Operations builds upon these historical insights, offering a consistent framework to manage negatives, roots, and exponents without ambiguity. By reinterpreting these operations within a unified system, the Canonical Order simplifies computation and demystifies concepts previously associated with complex numbers.

The Misconceptions of the Standard Order and the Introduction of the Canonical Order

The First Misconception

A negative times a negative equals a positive. While technically true, it is actually a negative cancels a negative and there must be two negatives already present. This concept is often misapplied. The root cause lies in efforts to simplify exponentials by framing them as repeated multiplication, such as:

$$x^n = x \times x \times \ldots \text{ (n times)}$$

While again technically accurate, many stop there, and this explanation fosters a conflation of multiplication and exponentiation, leading to misunderstandings about the exponential operator.

The Second Misconception

Operators act independently of other operators. Many misunderstand the role of the negative sign in expressions like -5^2, *incorrectly* treating the negative as part of the base or assuming it interacts with the exponent. This confusion also extends to roots, where $\sqrt{x}$ is underemphasized as a fractional exponent.

A deeper logical fallacy arises when attempting to extend negative bases into fractional exponents. For example:

$$(-5)^5 = -3125 \; and \; (-5)^6 = 15625$$

But what would $(-5)^{5.5}$ equal? This is where some got confused and where arithmetic begins breaking into an ***imaginary realm*** of understanding, as fractional exponents involving negatives inherently introduce imaginary numbers:

$$(-5)^5 = \sqrt[5]{-3125^5}$$

This represents a break in practical arithmetic and shifts into theoretical constructs like complex numbers, which are often presented as if they are "real" when their interpretation is not intuitive or tied to the physical world. (-3125^5)^(1/5) should cancel out, but instead break down into an imaginary number depending on interpretation.

The Third Misconception

The expression $-5^2 = -25$ confuses many due to failings in the education system. The Standard Order of Operations resolves this as:

$$(-5)^2 = 25$$

This is interpreted as $(-5)(-5) = 25$. However, this conflates multiplication and exponentiation, leading to ambiguity that the Canonical framework eliminates.

The Fourth Misconception

The Order of Operations requires that parentheses must always be resolved first, yet the Standard Order breaks this requirement, which is misleading. For example, in $(-5)^2$:

- Parentheses group the base, -5, for the exponentiation operation.
- The combination of multiplication $((-5)(-5))$ with exponentiation is often how $(-5)^2$ is misinterpreted.

Exponentiation inherently acts on the base, independent of surrounding operators, as it affects only the leftmost number. Correct resolution ensures only one base is acted upon, provided the base does not contain variables.

The Fifth Misconception

Negative signs are unaffected by other operators. For example:

$$-5 \times 6 = -30 \; not \; 30$$

$$(-5) \times 6 = -30 \; not \; 30$$

The Sixth Misconception

If a result cannot be inverted like its incremental counterparts, it is inherently flawed. For instance:

$+1, +2, +3$ can be undone with $-1, -2, -3$, ensuring mathematical consistency.

However, consider:

$$(-2)^2 = 4 \ (-2)^3 = -8 \ (-2)^4 = 16$$

These results cannot be consistently undone using operations such as $\sqrt{4}$, $\sqrt[3]{-8}$, or $\sqrt[4]{16}$, revealing flaws in the Standard Order.

The Canonical Order

The **Canonical Order** eliminates ambiguities caused by symbols like the square root, which introduces errors across mathematical literature due to inconsistent sign placement interpretations. For example:

$$-\sqrt{-5} \ simplifies \ to \ -(-5^{\frac{1}{2}})$$

Such ambiguities create deeper errors as complexity increases

The Canonical Order prioritizes understanding over aesthetics

$$x^{\frac{2}{3}} = the \ cube \ root \ of \ x^2 \ or \ \sqrt[3]{x^2}$$

$$x^{\frac{3}{4}} = the \ fourth \ root \ of \ x^3 \ or \ \sqrt[4]{x^3}$$

Identifying Biases in the Standard Order

The First Bias

In the Standard Order, we treat $(-5)^2$ as:

$$(-5)^2 = (-5)(-5) = 25$$

However, we do not treat $(-5) \times 2$ in the same way, instead solving as:

$$(-5) \times 2 = 0 - (5 \times 2) = -10$$

The Second Bias

In the Standard Order:

$$(5^2)^4 = 5^{2\times4} = 5^8 = 390625$$

But introducing a negative base creates inconsistent logic:

$$(-5^2)^4 = (-25)^4 = (-25)(-25)(-25)(-25) = 390625$$

Under the Canonical Order, such inconsistencies are discarded. Instead, operations are performed consistently and logically:

$$(-5^2)^4 = -5^{2\times4} = -5^8 = -390625$$

$$(-5)^4 = (-5^1)^4 = -5^{1\times4} = -625$$

Rewriting Roots as Powers (Canonical Order)

Single Roots

$$\sqrt{a} = a^{\frac{1}{2}}$$
$$\sqrt[3]{a} = a^{\frac{1}{3}}$$
$$\sqrt[4]{a} = a^{\frac{1}{4}}$$
$$\sqrt[n]{a} = a^{\frac{1}{n}}$$

Roots with Exponents

$$\sqrt{a^m} = a^{\frac{m}{2}}$$
$$\sqrt[3]{a^m} = a^{\frac{m}{3}}$$
$$\sqrt[4]{a^m} = a^{\frac{m}{4}}$$
$$\sqrt[n]{a^m} = a^{\frac{m}{n}}$$

Parentheses and Power Distribution

$$(\sqrt{a})^m = (a^{\frac{1}{2}})^m = a^{\frac{m}{2}}$$
$$(\sqrt[3]{a})^m = (a^{\frac{1}{3}})^m = a^{\frac{m}{3}}$$
$$(\sqrt[n]{a})^m = (a^{\frac{1}{n}})^m = a^{\frac{m}{n}}$$

Nested Roots

$$\sqrt{\sqrt{a}} = \sqrt[4]{a} = a^{\frac{1}{4}}$$
$$\sqrt{\sqrt[3]{a}} = \sqrt[6]{a} = a^{\frac{1}{6}}$$
$$\sqrt[3]{\sqrt{a}} = \sqrt[6]{a} = a^{\frac{1}{6}}$$
$$\sqrt[3]{\sqrt[3]{a}} = \sqrt[9]{a} = a^{\frac{1}{9}}$$

Negative Roots

$$-\sqrt{a} = -a^{\frac{1}{2}}$$
$$-\sqrt[3]{a} = -a^{\frac{1}{3}}$$
$$-\sqrt[n]{a^m} = -a^{\frac{m}{n}}$$

Reciprocal Roots

$$\frac{1}{\sqrt{a}} = a^{-\frac{1}{2}}$$
$$\frac{1}{\sqrt[3]{a}} = a^{-\frac{1}{3}}$$
$$\frac{1}{\sqrt[n]{a}} = a^{-\frac{1}{n}}$$
$$\frac{1}{\sqrt[n]{a^m}} = a^{-\frac{m}{n}}$$

Negative Reciprocal Roots

$$-\frac{1}{\sqrt{a}} = -a^{-\frac{1}{2}}$$
$$-\frac{1}{\sqrt[3]{a}} = -a^{-\frac{1}{3}}$$
$$-\frac{1}{\sqrt[n]{a}} = -a^{-\frac{1}{n}}$$

Combined Power and Root

$$\sqrt[n]{a^m} = a^{\frac{m}{n}}$$
$$(\sqrt[n]{a})^m = (a^{\frac{1}{n}})^m = a^{\frac{m}{n}}$$

*** Section subject to refinement**

I. **Implicit Power of 1 for Any Number or Variable**

$$a = a^1$$

Every number or variable is always implicitly raised to the power of 1, even if not explicitly written. This foundational rule enables consistent application of exponent laws, establishing each base as a^1

II. **Multiplying Powers with the Same Base**

$$a^m \times a^n = a^{m+n}$$

When multiplying expressions with the same base, add the exponents. This rule consolidates powers correctly: $a^2 \times a^3 = a^{2+3} = a^5$

III. **Dividing Powers with the Same Base**

$$\frac{a^m}{a^n} = a^{m-n} \quad (\text{where } a \neq 0)$$

When dividing expressions with the same base, subtract the exponent of the denominator from that of the numerator. This ensures accurate reduction: $\frac{a^5}{a^2} = a^{5-2} = a^3$

IV. **Power of a Power**

$$(a^m)^n = a^{m \times n} = a^{mn}$$

When an exponentiated expression is raised to another power, multiply the exponents. Following the order of operations, parentheses must be removed first, achieved here by applying this rule: $(a^2)^3 = a^{2 \times 3} = a^6$

V. **Parenthesized Base Raised to a Power**

$$(a)^m = (a^1)^m = a^{1 \times m} = a^m$$

For a variable or number in parentheses, recognize the base as implicitly raised to the power of 1. According to standard arithmetic rules, each number or variable is treated as a^1, which enables removal of parentheses and application of the power: $(a)^m = (a^1)^m = a^{1 \times m} = a^m$

VI. **Zero Exponent Law**

$$a^0 = 1 \quad (\text{where } a \neq 0)$$

Any non-zero number raised to the power of zero is always equal to 1. This universal law applies to all bases: $5^0 = 1$

VII. **Negative Exponent Law**

$$a^{-m} = \frac{1}{a^m} \quad (\text{where } a \neq 0)$$

A negative exponent inverts the base, resulting in the reciprocal of the base raised to the positive exponent: $a^{-2} = \frac{1}{a^2}$.

VIII. **Distributing a Power over Multiplication**

$$(ab)^m = a^m \times b^m$$

When a product is raised to a power, distribute the exponent to each factor within the parentheses. This step is necessary for accurate results: $(2x)^3 = 2^3 \times x^3 = 8x^3$

IX. **Distributing a Power over Division**

$$\left(\frac{a}{b}\right)^m = \frac{a^m}{b^m} \quad (\text{where } b \neq 0)$$

When a fraction is raised to a power, distribute the exponent to both numerator and denominator. This ensures proper handling of fractional powers: $\left(\frac{2}{3}\right)^2 = \frac{2^2}{3^2} = \frac{4}{9}$

X. **Power as the Inverse (Root Interpretation)**

$$a^{\frac{1}{n}} = \sqrt[n]{a}$$

Raising a number to the power of $\frac{1}{n}$ equates to taking the nth root of that number: $a^{\frac{1}{2}} = \sqrt{a}$.

XI. **Fractional Exponents Representing Power and Root**

$$a^{\frac{m}{n}} = \sqrt[n]{a^m}$$

A fractional exponent implies both a power and a root, with the denominator as the root and the numerator as the power. This interpretation is essential: $a^{\frac{3}{2}} = \sqrt{a^3}$.

XII. **Power of 1**

$$a^1 = a$$

Any number raised to the power of 1 remains unchanged. This rule is absolute: $7^1 = 7$.

XIII. **Zero to a Positive Exponent**

$$0^m = 0 \quad (\text{for } m > 0)$$

Zero raised to any positive exponent is always zero, without exception: $0^5 = 0$.

XIV. **Interpreting Negative Signs with Powers**

For expressions with negative bases raised to powers, the negative sign remains external when the base is implicitly raised to the power of 1. Applying the implicit unity concept allows consistent handling of negatives:

$$(-a)^m = (-a^1)^m = -a^{1 \times m} = -a^m$$

This interpretation maintains the negative outside the exponentiated term, ensuring proper evaluation.

The Law of Implicit Unity in Exponential Powers

This law establishes that every base, whether explicitly stated or in parentheses, is implicitly raised to the power of 1, enabling the removal of parentheses without altering the expression. It reinforces that parentheses containing bases should be viewed as a^1, ensuring consistent exponentiation:

1. For any term in parentheses, the base can be considered raised to the implicit power of 1.

   $$(a)^m = (a^1)^m = a^{1\times m} = a^m$$

2. In cases involving negatives, such as $(-a)^m$, apply this implicit unity to the absolute base:

   $$(-a)^m = (-a^1)^m = -a^{1\times m} = -a^m$$

Framework Overview

Unified Rules for Roots and Exponents

The Canonical Order establishes explicit rules for handling roots and exponents across all base values, including negatives. Fractional exponents consistently replace the radical symbol, removing ambiguities and ensuring compatibility with the real number system.

Key examples include:

- $\sqrt{4} = 4^{\frac{1}{2}}$
- $\sqrt[3]{8} = 8^{\frac{1}{3}}$
- $\sqrt{-9}$ reinterpreted as $(-9)^{\frac{1}{2}}$, retaining its real-number definition.

This framework eliminates inconsistencies, providing learners with a single, coherent system.

Explicit Treatment of Negative Bases

Negative bases are explicitly incorporated into the rules governing exponents, eliminating the need for imaginary constructs. For instance, operations such as $(-3)^{\frac{2}{3}}$ are clarified using consistent principles that naturally extend from the treatment of positive bases.

Elimination of the Radical Symbol

The radical symbol ($\sqrt{}$), a source of confusion for many learners, is entirely replaced by fractional exponents. This substitution aligns all operations under a unified framework, bridging the gap between basic arithmetic and advanced mathematical concepts.

Educational Advantages

Simplification of Mathematical Concepts

The elimination of symbols such as $\sqrt{}$ and the consistent treatment of operations reduce cognitive load for students. Learners can more easily grasp fractional exponents when negative bases are treated systematically.

For example, operations such as $\sqrt[3]{-8} = (-8)^{\frac{1}{3}}$ are rendered intuitive, fostering a deeper understanding of mathematical principles.

Improved Transition to Advanced Topics

By clarifying foundational concepts, the Canonical Order facilitates a smoother progression from arithmetic to algebra, calculus, and beyond. Students encounter fewer conceptual hurdles, improving their confidence and engagement with mathematics.

Conclusion

The Canonical Order of Operations provides a transformative framework for mathematics, addressing long-standing ambiguities in the Standard Order. By treating negatives consistently, redefining roots as fractional exponents, and eliminating the radical symbol, this system simplifies computation and enhances learning.

This framework realizes the historical aspirations of mathematicians who questioned the necessity of imaginary numbers, offering a unified approach that retains the integrity of the real number system. With its emphasis on clarity and consistency, the Canonical Order equips learners and practitioners with a robust, intuitive foundation for mathematical exploration.

Resources

Online Calculator Demo: https://canonical.streamlit.app
Github Python Code: https://github.com/andylehti/canonical-order

```
import time

def normEx(e):
    return e.replace(' ', '').replace('^', '**')

def nestEx(e):
    s = []
    for i, c in enumerate(e):
        if c == '(':
            s.append(i)
        elif c == ')':
            if s:
                return s.pop(), i
            else:
                return -1, -1
    return -1, -1

def simpEx(e):
    e = normEx(e)
    x = [e]
    while '(' in e:
        a, b = nestEx(e)
        if a == -1 or b == -1:
            return ["Error: Unmatched parentheses."]
        try:
            # Safely evaluate the expression within the parentheses
            r = str(eval(e[a + 1:b]))
        except Exception as ex:
            return [f"Error evaluating '{e[a + 1:b]}': {ex}"]
        e = e[:a] + r + e[b + 1:]
        x.append(e)
    try:
        # Evaluate the final expression
        final_result = str(eval(e))
        x.append(f"= {final_result}")
    except Exception as ex:
        x.append(f"Error evaluating final expression '{e}': {ex}")
    return x

def calcEx(e):
    print(f"Original Expression: {e}\n")
    steps = simpEx(e)
    print("Simplification Steps:")
    for step in steps:
        print(step)

input_expression = "(-25)^2"
calcEx(input_expression)

        # Note: This is experimental code, and may not work in all cases.
```